Brief Announcement: Distributed Broadcasting and Mapping Protocols in Directed Anonymous Networks*

Michael Langberg
The Open University of Israel
Computer Science Division
Raanana 43107, Israel
mikel@openu.ac.il

Moshe Schwartz
Ben-Gurion University
Department of Electrical and Computer Engineering
Beer Sheva 84105, Israel
schwartz@ee.bgu.ac.il

Jehoshua Bruck
California Institute of Technology
Pasadena, CA 91125, USA
bruck@paradise.caltech.edu

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols, Distributed Systems

General Terms

Algorithms, Theory

Keywords

Anonymous networks, directed networks, broadcast

1. INTRODUCTION

In this work we study the fundamental problems of broadcasting and mapping (label assignment and topology extraction) in directed anonymous networks. In such a network G, processors do not have unique identifiers, they execute identical protocols, and they have no knowledge of the topology of the network (even the size or bounds on it are unknown). The only knowledge available to a vertex is its own degree.

Anonymous and unknown networks have been extensively studied during the last few decades. These studies include graph exploration, where a robot has to construct a complete map of an unknown environment; the study of network communication or more specifically the task of broadcasting and label assignment; and various characterizations of attainable and unattainable tasks with respect to additional symmetry breaking assumptions. Due to space limitations, a detailed reference list is omitted and appears in the full version of the paper*.

All studies above consider the underlying network to be undirected or directed but strongly connected. In such networks, the typical paradigm used in the context of anonymous networks is adaptive message passing. Namely, vertices in the network learn of each other via information transmitted up an down paths that connect these vertices, and construct their outgoing messages based on the information gained so far. In this work we address the design of distributed protocols when the underlying network is **directed but not necessarily strongly connected**. This setting differs substantially from the undirected case as a vertex may only have a single chance to transmit an outgoing message, and thus the protocols must be designed accordingly.

* A full version of this paper is available at `http://www.paradise.caltech.edu/papers/etr084.pdf`

PODC'07, August 12–15, 2007, Portland, Oregon, USA.
ACM 978-1-59593-616-5/07/0008.

2. OUR CONTRIBUTION

This work initiates the study of distributed asynchronous protocols over directed anonymous networks that are not necessarily strongly connected. We study the model is which the given network G has two *special* vertices: a root s and a terminal t (such that there is a path connecting s and t). Our access to the network G is obtained solely through these two vertices and our objective is to preform certain tasks on G. Roughly speaking, the protocols we present proceed as follows. An initial message is sent from the root vertex s to its children. This initiates the distributed protocol which ends when the terminal vertex is in a final state with its state as output.

This model of study, in which we are to access an unknown network G through the special vertices s and t, corresponds, for example, to the scenario in which maintenance operations are to be preformed on a dynamically growing network G of unknown topology. In this scenario, as long as one maintains a port in which messages can be inserted to the network (*i.e.* the node s) and a port in which messages can be retracted form the network (*i.e.* the node t), the maintenance operation may be performed (triggered from s) and once completed a status report can be obtained at t.

The algorithmic tasks we present in this work are the basic tasks of broadcasting and label assignment. For broadcasting, a message m is given at the root s, and we wish to distribute m throughout the entire network G. This, in itself, seems a trivial task obtained by simple propagation of m. However, we also want the protocol to terminate iff all the vertices of G have received m. This turns out to be significantly more involved as our protocol cannot use the standard termination techniques used for graphs which are strongly connected (namely, message passing).

To this end, each internal vertex that has received m will send out an additional message or messages that eventually will reach the terminal vertex t. These additional massages should be constructed in such a way that t will be able to decide whether all vertices of G have been visited or not. In a nutshell, the additional information sent will represent a certain commodity, and the actions taken by each vertex in G will be *commodity-preserving* (for example each vertex can partition its incoming commodity among its outgoing edges). Thus, if the source were to send a unit of commodity into the graph G and over time the terminal were to receive a unit of commodity, one could conclude that all vertices in G have been visited. Such ideas lend themselves naturally to acyclic graphs, and the main difficulty in our work lies in the case in which G contains cycles.

We start by presenting and analyzing a simple broadcasting protocol that will terminate correctly if G is essentially a tree. We then present the main result of this work, a distributed asynchronous broadcasting protocol for general directed graphs. Finally we turn to design a protocol which assigns unique labels to all vertices of G (thus enabling us to map the graph topology) and terminates after these labels have been assigned. All the protocols we present are distributed and asynchronous. We consider several measures of complexity: total communication complexity – the number of bits transmitted throughout the protocol, required bandwidth – maximal number of bits transmitted over a single edge, and the maximal number of bits in a label. Our model and results can be roughly summarized as follows.

3. THE MODEL

In this work we study anonymous protocols on directed graphs $G = (V, E)$ with two special vertices. The root vertex, denoted by s and the terminal vertex denoted by t. Vertices in $V \setminus \{s, t\}$ will be referred to as internal vertices. We assume that s has no incoming edges and only one outgoing edge, and t has no outgoing edges. As our access to G is only through s and t, if there are internal vertices which are not on a path from s to t but are still reachable from s or connected to t, our protocols will not terminate.

All vertices in G are assumed to know nothing of the topology of the network (including its size) nor do they have unique identifiers. Each vertex is assumed to know how many incoming and outgoing edges it has, and has the power to distinguish between different incoming/outgoing edges. The model we present is asynchronous. Our results can be easily extended to the case in which there are multiple root/terminal vertices, the root has multiple outgoing edges, and to the case that the communication throughout the network is synchronous.

Anonymous protocols: An anonymous protocol on G is defined by the following primitives: a state space Π, a message space Σ, an initial state $\pi_0 \in \Pi$, an initial massage σ_0, a state function $f : \Pi \times \Sigma \times \mathbb{N} \to \Pi$, a message function $g : \Pi \times \Sigma \times \mathbb{N} \times \mathbb{N} \to \{\Sigma, \phi\}$, and a stopping predicate $S : \Pi \to \{0, 1\}$.

An anonymous protocol is executed as follows. Initially we associate the state π_0 with every vertex in G. The message σ_0 is sent on the outgoing edge of s. Each vertex that receives message σ on its i-th incoming edge while having current state π, moves to state $\pi' = f(\pi, \sigma, i)$ and sends message $g(\pi, \sigma, i, j)$ on its j-th outgoing edge. If in the scenario above $g(\pi, \sigma, i, j) = \phi$, then no message is sent on outgoing edge j. We say that an anonymous protocol has terminated if $S(\pi) = 1$ for the state π of t, in this case π is the output of the protocol.

Quality: There are several quality parameters that may be considered when studying anonymous protocols. The size of the state space is related to the amount of memory needed at each vertex of the network. The size of the message space is a bound on the maximum message length transmitted on edges on the network (i.e., bandwidth). Multiplying the bandwidth by the total number of messages sent over the network before termination will imply an upper bound on the total communication complexity of the protocol. In a synchronous model one may also consider the time it takes for the protocol to terminate.

In this work we focus on the asynchronous model in which we seek to design anonymous protocols with minimal total communication complexity. To this end, we study both the total number of messages transmitted throughout the network, and the maximum message size. There is an obvious trade-off between the two, and their product is the total communication complexity.

4. OUR RESULTS

Broadcasting in acyclic networks: A graph G is said to be a *grounded tree* if every vertex of G has in-degree 1, excluding the source s with no incoming edges and the terminal t which may have several incoming edges.

For grounded trees G, we describe an asynchronous distributed protocol that broadcasts a message m from the root vertex s to all of G, and halts iff all vertices have received m. In what follows, we denote the size of m by $|m|$. We show that our results are tight by providing a matching lower bound on the total communication complexity of the suggested protocol.

THEOREM 4.1 (UPPER AND LOWER BOUND). *The total communication complexity for broadcasting in grounded trees is* $\Theta(|E| \log |E|) + |E|\,|m|$.

For general directed acyclic graphs we also provide an asynchronous distributed protocol for broadcasting. Our results are the best possible when considering certain protocols we refer to as *commodity preserving*.

THEOREM 4.2. *The total communication complexity for broadcasting in directed acyclic graphs is* $O(|E|^2) + |E|\,|m|$.

Broadcasting in general networks: For general G, we describe an asynchronous distributed protocol that broadcasts a message m from the root vertex s to all of G, and halts iff all vertices have received m. In what follows, let d_{out} be the maximal out-degree in the given network G.

THEOREM 4.3. *The total communication complexity for broadcasting in general networks is* $O(|E|^2\,|V| \log d_{\text{out}}) + |E|\,|m|$.

Unique label assignment: For general G, we describe an asynchronous distributed protocol that assigns unique labels to all vertices of G, and halts iff all vertices have been assigned labels.

THEOREM 4.4. *The total communication complexity for label assignment in general networks is* $O(|E|^2\,|V| \log d_{\text{out}})$*. Each resulting vertex label is of length* $O(|V| \log d_{\text{out}})$ *bits.*

The resulting label complexity stated above is surprisingly high. One might expect to be able to label the vertices with at most $O(\log |V|)$ bits, which is achievable in both the undirected case and the directed but strongly-connected case. However, we show our result to be tight, and the exponential blowup necessary.

THEOREM 4.5. *In the directed anonymous setting, for any unique labeling protocol there exist infinitely many networks* $G = (V, E)$ *for which the protocol will produce labels of length* $\Omega(|V| \log d_{\text{out}})$.